

CONTROLE DE TEMPERATURA COM ARDUÍNO

Michelle Silva Lima – michelleisislima@gmail.com
Rayane Fontanella Santiago – rayfsantiago@hotmail.com
João Pedro Maciel Trajano – j.pedromec@hotmail.com
Leonardo Chahim Pereira – chahim@outlook.com.br
Sanderson Rocha de Abreu – sanderson.abreu@unis.edu.br

RESUMO: O presente artigo aborda a visualização e controle de temperatura de uma caixa térmica aplicando a tecnologia de prototipagem Arduíno. Tal abordagem se faz necessária devida ao fato de caixas térmicas serem utilizadas no transporte de vacinas, medicamentos, alimentos e outras aplicações nas quais é necessário o controle da temperatura interna da caixa. As normas esclarecem que as vacinas devem ser conservadas a temperaturas entre 2°C e 8°C, assim sendo, foi ajustado no controlador um alarme sonoro para temperaturas abaixo e acima desta faixa, e utilizado o display para monitoramento em tempo real das variações de temperatura, conseguindo assim um controle baseado nos princípios de Poka Yoke. As temperaturas são controladas pela adição de gelo ou abertura da tampa da caixa para abaixar ou aumentar a temperatura interna. Para transporte de vacinas, o controle de temperatura da caixa térmica possibilitou um acondicionamento adequado para manutenção de suas propriedades. Introduzindo o sensor dentro de um isopor simulando uma geladeira, o projeto foi configurado para que fosse monitorada a temperatura, de modo que não passasse de 8°C, pois, acima disso, a vida útil das vacinas estaria em risco.

Palavras-chave: Temperatura. Arduíno. Conservação.

1 INTRODUÇÃO

Os medicamentos, assim como outros produtos, necessitam de cuidados para que conservem suas propriedades físicas e químicas. E um dos fatores responsáveis pelo resguardo dessas propriedades, mantendo íntegro seus princípios ativos é a devida preservação, o que engloba formas de armazenar e a temperatura a que estão submetidas.

Nas áreas hospitalares, clínicas e farmacêuticas, a má preservação dos fármacos é um dos fatores que mais influencia para a perda de vacinas. Em consequência, a empresa ou instituição pode ser autuada pela Agência Nacional de Vigilância Sanitária (ANVISA), além de ter ônus financeiros. Essas providências devem ser tomadas devido à alta responsabilidade, pois coloca em risco a segurança dos pacientes.

O objetivo do artigo é construir um protótipo de monitoramento de temperatura para o armazenamento de vacinas com a instalação do *Led* (*light-emitting diode*) verde e vermelho, e um *buzzer* indicando para o usuário se a temperatura está acima ou abaixo do especificado. Fazendo assim, o equipamento funcionar com princípio de *Poka Yoke*.

Este método, que foi desenvolvido no Japão e significa “à prova de erros”, criada nos anos 1960 e implantada no Sistema Toyota de Produção, a função deste método é corrigir falhas e erros em processos industriais e evitar falhas humanas, sempre por meio de ações simples.

Os princípios devem ser empregados desde o início, até o final do processo, e é baseado em princípios à prova de erros, logo, quando um erro é detectado, ele não se transforma em um

defeito e assim, pode-se eliminar as suas causas. Um exemplo simples é a impossibilidade de remover a chave quando está na ignição do seu automóvel se a sua transmissão automática estiver em outra posição que não seja o “ponto morto”, impossibilitando ao dono do veículo de sair em condições inseguras.

As próximas seções do artigo serão apresentadas o protótipo de controle de temperatura em uma caixa térmica, usando o princípio de *Poka Yoke* com objetivo de evitar perdas.

2 REVISÃO DE LITERATURA

De acordo com MARTINAZZO *et al* (2014), o *Arduino* é uma peça que pode auxiliar o estudo em escolas e universidades, devido à capacidade de coleta de dados e sua possibilidade de inclusão com sensores, dado que essa plataforma possibilita análises de fenômenos diversos, tais como resistência, corrente elétrica e capacitância, além do conhecimento sobre microcontroladores e processadores.

O *Arduino* tem como a base o microcontrolador AVR, da qual a programação é executada num computador e em seguida passada para o *Arduino* via cabo USB; é uma ferramenta de código aberto, de linguagem C/C++ (MARTINAZZO, 2014), utilizada para “obtenção de dados de sensores de entrada e de saída” (Martinazzo *et al*, 2014, p. 24).

Em definições práticas, um *Arduino* é um microcomputador que você pode programar para organizar entradas e saídas entre a ferramenta e os itens que estejam ligados a ele externamente (MCROBERTS, 2011).

Os dispositivos a prova de erro (*poka-yoke*) são designados a impossibilitar a ocorrência de falhas na produção. O *poka-yoke* pode ser usado para reparar erros em dois modos: quando ativado na produção, de forma em que o problema pode ser reparado no momento; e o método de advertência, quando é ativado, um alarme toca ou uma luz sinaliza, atentando o trabalhador (SHINGO, 1996).

Poka-yoke é um dispositivo de identificação de irregularidades, um sistema de inspeção de erros. O conceito de *poka-yoke* em japonês é prevenir ou evitar erro. Pode ser definida como uma ferramenta à prova de erros, construída para impedir erros básicos do ser humano (CONSUL, 2015).

O serviço de vacinação aplica métodos para conservar a qualidade e as vacinas, caso tenham que ser transportadas. Na seção IV da Resolução da Diretoria Colegiada, na gestão da tecnologia e das técnicas parágrafo 2º, a condição de temperatura durante o trajeto deve ser controlada juntamente ao registro das temperaturas mínima e máxima (ANVISA, 2017).

(Oliveira, p 3, 2009) O PNI - Programa Nacional de Imunização, estabelece como rede de frio a união de ligação entre as posições nacional, estadual, regional, municipal e local, incluindo o transporte, a manipulação e o armazenamento dos imunobiológicos em exigências apropriadas. As vacinas são armazenadas e conservadas em temperaturas específicas, considerando sua

composição. Na posição nacional, alguns imunobiológicos são conservados em temperaturas negativas, já no local são refrigeradas entre 2° C a 8° C, em refrigeradores específicos e próprios.

3 MÉTODOS E MATERIAIS

Para a elaboração do presente artigo científico utilizou-se os critérios de pesquisa básica através de artigos pesquisados no Google Acadêmico e no Scielo (Biblioteca Eletrônica Científica Online), sites como da ANVISA (Agência Nacional de Vigilância Sanitária), por meio de assuntos relacionados a controle de temperatura com *Arduíno* e equipamentos para a conservação das vacinas. Com abordagem no embasamento teórico, que se refere aos assuntos que irão ser abordados no presente artigo é determinada o modo de pesquisa a ser aplicada no protótipo, o objetivo da pesquisa, a forma de abordar o problema e os procedimentos técnicos adotados no controle de temperatura.

3.1 Montagem da caixa térmica (protótipo)

Primeiramente, desconecta-se o *Arduíno* do cabo USB para se ter certeza que este esteja desligado. Após esta etapa, conectam-se os *Leds* (*light-emitting diode*) e o *buzzer* para sinalização de quando a temperatura atingir o ponto crítico programado, o Sensor de Temperatura DS18B20 à Prova D'Água, os fios que transmitam o que o código manda para cada componente, os potenciômetros e o LCD (*Liquid Crystal Display*).

Neste protótipo, foi optado por utilizar fios com cores diferentes, mas as cores não importam, desde que não haja incerteza durante a montagem na *proto-board*, e que os componentes e os fios estejam conectados na ordem correta. A inserção incorreta ou descuidada dos componentes na *proto-board* pode danificá-la, por isso, foi necessário ter um cuidado extra para que nenhum erro ocorresse. Após isso, conecta-se o cabo USB ao computador e liga-se o *Arduíno* para inserção do código do projeto.

Para a elaboração do Controle de Temperatura com o *Arduíno*, foram utilizadas as ferramentas de acordo com a Tabela 1.

Tabela 1: Ferramentas utilizadas

| Ferramentas utilizadas |
|---|
| Sensor de Temperatura DS18B20 à Prova D'Água |
| Fios <i>Jumpers</i> |
| LCD (<i>Liquid Crystal Display</i>) - JHD 162A de 16 colunas e 2 linhas |
| Potenciômetros de 10K ohms |
| <i>Proto-board</i> |
| <i>Arduíno</i> Uno R3 |
| <i>Leds</i> (<i>light-emitting diode</i>) vermelho e verde |
| <i>Buzzer</i> |
| Isopor 5 litros |

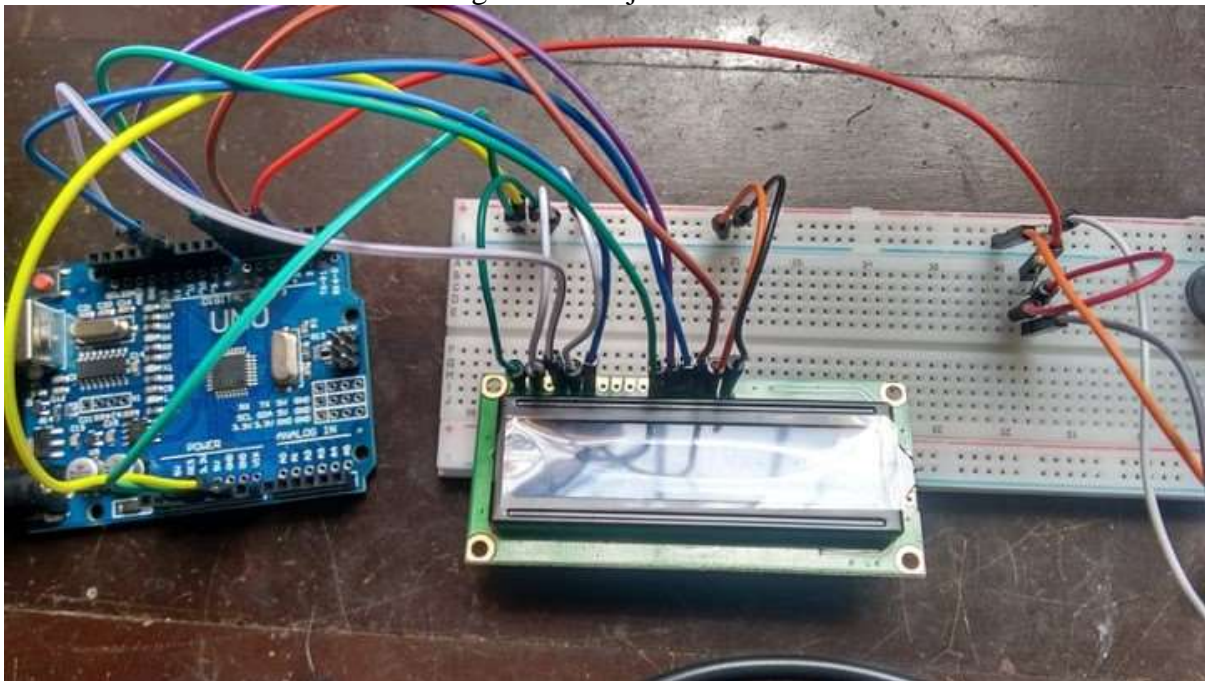
Fonte: Os Autores (2019)

Logo após a montagem de todos os seus componentes, é conectado com um cabo USB para que se possa rodar o código do programa (anexo 1).

3.3 Procedimentos experimentais

O protótipo desenvolvido foi montado com as ferramentas e os métodos utilizados a para elaboração do Controle de Temperatura com o *Arduíno*, de acordo com a figura 1.

Figura 1 – Projeto Montado



Fonte: Os Autores (2019)

Com o projeto montado como mostra a figura 1, o Sensor de Temperatura DS18B20 à Prova D'Água fica acoplado dentro do isopor para que ele nos mostre a temperatura para o controle das vacinas. Quando a tampa do isopor se abre, pela lei da Termodinâmica as temperaturas diferentes tendem a trocar calor para que ambos estejam em equilíbrio, com isso o isopor começa a trocar calor com o ambiente, e quanto mais tempo se passa, maior é a temperatura dentro do isopor. As vacinas podem perder sua validade, podendo assim, ser um problema e risco na saúde das pessoas. Depois de colocados os *Leds* (*light-emitting diode*) sinalizadores foi projetada temperatura ideal e temperatura crítica, o *LED* verde ligado, significa não haver problemas, porém, ao passar da temperatura crítica, aciona-se o *LED* vermelho para que a pessoa possa fechar o isopor e evitar que as vacinas se percam.

4 RESULTADOS E DISCUSSÕES

Com o projeto finalizado, com suas conexões ao LCD (*Liquid Crystal Display*), no *Arduíno*, *Leds* (*light-emitting diode*), *buzzer*, sensor de temperatura e caixa de isopor, foi compilado o código, como mostra a figura 2.

Figura 2 – Projeto concluído



Fonte: Os Autores (2019)

Com todos os componentes ligados e com o código compilado pelo software *Arduíno*, o sensor foi introduzido dentro do isopor, como mostra a figura 2. O projeto foi configurado para que fosse monitorada a temperatura e não passasse de 8°C, pois, acima disso, a vida útil das vacinas estaria em risco, o que consequentemente colocaria também a vida das pessoas. Então, foi testado com a tampa aberta e foi notado que depois de alguns minutos, a temperatura começou a subir devido à troca de calor com o ambiente externo, e a temperatura estipulada logo então é excedida. Foi feito uso de um *LED* (*light-emitting diode*) vermelho com a finalidade de sinalizar ao usuário que a temperatura alcançou o valor crítico (no caso 8°C), e um *buzzer* a fim de adicionar um efeito sonoro ao alarme, para que a pessoa responsável possa ver e ouvir os sinais, e assim fechar o isopor para que a temperatura no interior comece a se estabilizar. Quando estabilizado na temperatura ideal, o *LED* (*light-emitting diode*) verde então se acende.

Os resultados apresentados demonstram um sistema simples que funciona com o princípio de *Poke Yoke*, que foi planejado para poder salvar vidas, não ter desperdício de vacinas estragadas por descuido da falha humana e não ter perda de dinheiro. Sendo assim, evitar que seja necessário a compra de um novo lote de vacinas.

5 CONCLUSÃO

Ao final deste projeto, acredita-se que o controle de temperatura é necessário para que a perda de materiais e possivelmente de dinheiro sejam evitadas. Além de colocar a vida de seus pacientes em risco. Este propósito foi conseguido mediante estudo de caso onde se verificou a necessidade de limitar temperaturas para transporte de vacinas em caixas térmicas, para aplicação em locais afastados dos postos de atendimento médico, para conhecer as temperaturas adequadas para conservação das vacinas, realizou-se consultas às normas vigentes da ANVISA (Agência Nacional de Vigilância Sanitária).

Levando-se em consideração esses aspectos, o projeto teve como base o princípio básico de *Poka Yoke*, introduzido pelo Sistema Toyota de Produção, a fim de evitar falhas humanas através de ações simples, neste caso, a falha seria o responsável acidentalmente esquecer as vacinas fora do ambiente refrigerado. Foi feita a utilização de um meio sonoro (*buzzer*) e visual (*Led - light-emitting diode*) programados para que quando a temperatura máxima permitida for atingida (8°C), o usuário não cometa a falha, logo, quando o erro é detectado, ele não se tornará um defeito, podendo assim, eliminar a sua causa rapidamente.

REFERÊNCIAS BIBLIOGRÁFICAS

- ANVISA. **Agência Nacional de Vigilância Sanitária**. Disponível em:
http://portal.anvisa.gov.br/documents/10181/2718376/RDC_197_2015_.pdf/44ed78c4-1293-48f9-89f4-b89ad64cb27f Acesso em 20 de maio de 2019. CONSUL, J. T. Aplicação de *Poka-Yoke* em processos de caldeiraria. *Production*, v. 25, n. 3, p. 678- 690, jul./set. 2015. MARTINAZZO, C. A.; TRENTIN, D. S.; FERRARI, D.; PIAIA, M. M.; *Arduino*: Uma tecnologia no ensino de física. *Perspectiva*, Erechim. v. 38, n.143, p. 21-30, setembro/2014. MCROBERTS, Michael. *Arduino* Básico. [tradução Rafael Zanolli]. – São Paulo: Novatec Editora, 2011. 22 - 24 p.
- OLIVEIRA, Valéria Conceição de et al. Prática da enfermagem na conservação de vacinas. *Acta paul. enferm.*, São Paulo, v. 22, n. 6, p. 3-5, Dec. 2009. Disponível em:
http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-21002009000600014&lng=en&nrm=iso. Acesso em 22 de maio de 2019.
- SHINGO, S. O Sistema Toyota de produção. Porto Alegre: Artes Médicas, 199.

ANEXO 1:

Código utilizado no programa do *Arduíno* para o controle de temperatura.

Este código é utilizado para a medida de temperatura (escala Celsius) através do sensor DS18B20. Para carregar o código no programa e aparecer na tela de LCD (*Liquid Crystal Display*) é necessário antes se certificar que as bibliotecas *OneWire.h*, *DallasTemperature.h* e *LiquidCrystal.h* estão devidamente instaladas no *Arduíno*.

```
// Programa: Sensor de temperatura DS18B20
```

```
#include <OneWire.h>  
#include <DallasTemperature.h>  
#include <LiquidCrystal.h>
```

```
// Porta do pino de sinal do DS18B20  
#define ONE_WIRE_BUS 3
```

```
// Define uma instancia do oneWire para comunicacao com o sensor  
OneWire oneWire(ONE_WIRE_BUS);
```

```
// Armazena temperaturas minima e maxima
float tempMin = 999;
float tempMax = 0;

DallasTemperature sensors(&oneWire);
DeviceAddress sensor1;

// Inicializa o LCD
LiquidCrystal lcd(12, 11, 7, 6, 5, 4);

void setup(void)
{
  Serial.begin(9600);
  sensors.begin();
  // Localiza e mostra enderecos dos sensores
  Serial.println("Localizando sensores DS18B20...");
  Serial.print("Foram encontrados ");
  Serial.print(sensors.getDeviceCount(), DEC);
  Serial.println(" sensores.");
  if (!sensors.getAddress(sensor1, 0))
    Serial.println("Sensores nao encontrados !");
  // Mostra o endereco do sensor encontrado no barramento
  Serial.print("Endereco sensor: ");
  mostra_endereco_sensor(sensor1);
  Serial.println();
  Serial.println();
  lcd.begin(16, 2);
}

void mostra_endereco_sensor(DeviceAddress deviceAddress)
{
  for (uint8_t i = 0; i < 8; i++)
  {
    // Adiciona zeros se necessário
    if (deviceAddress[i] < 16) Serial.print("0");
    Serial.print(deviceAddress[i], HEX);
  }
}

void loop()
{
  // Le a informacao do sensor
  sensors.requestTemperatures();
  float tempC = sensors.getTempC(sensor1);
  // Atualiza temperaturas minima e maxima
  if (tempC < tempMin)
  {
    tempMin = tempC;
  }
  if (tempC > tempMax)
  {
    tempMax = tempC;
  }
}
```

```
}  
// Mostra dados no serial monitor  
Serial.print("Temp C: ");  
Serial.print(tempC);  
Serial.print(" Min : ");  
Serial.print(tempMin);  
Serial.print(" Max : ");  
Serial.println(tempMax);  
  
// Mostra dados no LCD  
lcd.clear();  
lcd.setCursor(0,0);  
lcd.print("Temp.:  ");  
//Simbolo grau  
lcd.write(223);  
lcd.print("C");  
lcd.setCursor(7,0);  
lcd.print(tempC);  
lcd.setCursor(0,1);  
lcd.print("L: "); }
```